UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/644,133	08/20/2003	Chris P. Karamatas	BEA920030013US1	5334

61780 7590 01/08/2008
LAW OFFICES OF MICHAEL DRYJA
1474 N COOPER RD #105-248
GILBERT, AZ 85233

EXAMINER
SPITTLE, MATTHEW D

ART UNIT	PAPER NUMBER
2111	

MAIL DATE	DELIVERY MODE
01/08/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

AK

Office Action Summary	**Application No.** 10/644,133	**Applicant(s)** KARAMATAS ET AL.	
	Examiner Matthew D. Spittle	**Art Unit** 2111	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 October 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,4-10,12-19,21-25 and 28-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4-10,12-19,21-25 and 28-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Claims 1, 4 – 10, 12 – 19, 21 – 25, and 28 – 30 have been examined.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1, 4 – 10, 12 – 19, and 21 – 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kiick (U.S. Pub. 2003/0200250) and what was well known in the art as evidenced by Rowlands et al. (U.S. 6,965,973), Fischer et al. (U.S. 6,438,672), Drottar et al. (U.S. 6,170,025), Agatsuma et al. (U.S. 7,237,099), and Chi et al. (U.S. 6,209,086).

Regarding claim 1, Kiick teaches a method comprising at least one of:

Assigning interrupts for a plurality of input/output (I/O) devices among a plurality of nodes of a system based on at least one of: the nodes to which the I/O devices are connected; the nodes at which interrupt service routines for the I/O devices reside; and processors of the nodes for the nodes having processors, where one or more of the nodes have processors and memory (Paragraph 34 describes that interrupts should be assigned to the "closest" processors, and not across node boundaries. Examiner interprets this to mean the interrupts for the I/O devices should be assigned to nodes to which they are connected or to nodes where the ISRs for the said I/O devices reside.);

For each node of the system having processors, assigning the interrupts for the devices that are performance critical and that have been assigned to the node among the processors of the node in a round-robin manner (Examiner interprets all devices in the reference to be considered "performance critical"; Paragraph 26);

Dynamically modifying assignments of the interrupts among the nodes of the system based on actual performance characteristics of the assignments (Paragraphs 25, 28, 31);

For each node of the system having processors, dynamically modifying assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node based on actual performance characteristics of the assignments (Paragraphs 25, 28, 31).

Kiick fails to teach where one or more of the nodes are memoryless.

Examiner takes Official Notice that nodes in a NUMA system may or may not contain memory. Rowlands et al. evidences this (col. 4, lines 1 – 2).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to provide memoryless nodes within the system of Kiick, since this is routine within NUMA systems.

Kiick fails to teach assigning the interrupts for the I/O devices to a first node having a cache, memory, and at least one processor and assigning the interrupt for the I/O device to the first node. Similarly, Kiick fails to teach repeating the assignment for the second and third nodes, respectively, if the first node to which the device is connected does not have a cache, memory, and at least one processor.

Examiner takes Official Notice that it would be obvious to assign the interrupt to a node which has a processor and memory, since these elements are required to service an interrupt (as evidenced by Agatsuma et al.; col. 4, line 64 - col. 5, line 9; col. 6, lines 15 - 22; Drottar et al.; col. 21, lines 15 - 22; col. 22, lines 44 - 58; col. 23, lines 22 – 30; and Fischer et al.; col. 9, lines 35 - 51). Examiner notes that all modern processor architectures contain a "memory" since they contain registers which are operated upon internally. Cache is old and notoriously well known in the art for providing improved performance in computing systems (as evidenced by Chi et al.; col. 2, lines 29 – 36; and Fischer et al.; col. 1, line 31 – col. 2, line 40).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to assign the interrupt for the I/O devices of Kiick to nodes which contain processors and memory, since both of these components are known in the prior art to be required in order to service an interrupt. It would have been obvious to use choose nodes which have cache, since it is well-established in this art that caching provides greater performance.

With regard to claim 4, Kiick teaches the method of claim 1, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises:

Measuring responsiveness of the node in processing the interrupt (paragraphs 27 – 30);

Kiick fails to explicitly teach assigning the interrupt to the node at which the interrupt service routine for the I/O device resides; measuring responsiveness of the node at which the interrupt service routine for the I/O device resides in processing the interrupt; and where the responsiveness of the node to which the I/O device is connected is better than the responsiveness of the node at which the interrupt service routine for the I/O device resides, reassigning the interrupt to the node to which the I/O device is connected.

Kiick does, however, teach that the dynamic interrupt distributor should be aware of the system architecture, and re-assign interrupts to the "closest" processors (paragraph 34). Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to give preference in assigning interrupts to nodes at which the ISR for the device resides, or at which the I/O device itself resides, and then to re-assign interrupts accordingly to which configuration produced better performance as described in paragraphs 28 – 30).

With regard to claim 5, Kiick teaches the method of claim 4, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises, for each assignment of an interrupt for an I/O device to a node, where the node is that at which the interrupt service routine for the I/O device resides:

Measuring responsiveness of the node in processing the interrupt (paragraphs 27 – 30);

Kiick fails to explicitly teach assigning the interrupt to the node to which the I/O device is connected; measuring responsiveness of the node to which the I/O device is connected in processing the interrupt; and where the responsiveness of the node at which the interrupt service routine for the I/O device is connected is better than the responsiveness of the node to which the I/O device is connected, reassigning the interrupt to the node at which the interrupt service routine for the I/O device resides.

Kiick does, however, teach that the dynamic interrupt distributor should be aware of the system architecture, and re-assign interrupts to the "closest" processors (paragraph 34). Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to give preference in assigning interrupts to nodes at which the I/O device is connected, or at which the ISR for the /O device itself resides, and then to re-assign interrupts accordingly to which configuration produced better performance as described in paragraphs 28 – 30).

With regard to claim 6, Kiick teaches the method of claim 1, wherein for each node of the system, dynamically modifying the assignments of the interrupts that are performance critical and that have been assigned to the node among the process ors of the node comprises:

Measuring the responsiveness of the processors of the node in processing the interrupts assigned thereto (paragraphs 27, 28, 35);

Where a differential between a best responsiveness and a worst responsiveness is greater than a threshold (paragraph 28; where a threshold may be interpreted as a "large enough difference");

Reassigning at least one of the interrupts assigned to the processor having the worst responsiveness to the processor having the best responsiveness (paragraphs 27 – 30, 35).

With regard to claim 7, Kiick teaches a non-uniform memory access (NUMA) system comprising:

A plurality of nodes (Figure 1, items 102A, 102B);

A plurality of input/output (I/O) devices, each I/O device connected to one of the plurality of nodes and having an interrupt (Figure 1, items 110A, 110B);

An interrupt-assignor responsive to the I/O devices and the nodes to assign the interrupt for each I/O device to one of the plurality of nodes in a performance-optimized manner (where an interrupt-assignor may be interpreted as a dynamic interrupt distributor; Figure 2, item 210; paragraphs 25, 28).

Kiick teaches a multiprocessor system that is tightly-coupled, and could have shared main memory, mass storage, and cache, and runs a single copy of an operating system (paragraph 7). These limitations define a NUMA system as evidenced by the definition fromt5 Whatis.com, and therefore, Kiick implicitly describes a NUMA system for use with his invention.

Kiick fails to teach where one or more of the nodes are memoryless.

Examiner takes Official Notice that nodes in a NUMA system may or may not contain memory. Rowlands et al. evidences this (col. 4, lines 1 – 2).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to provide memoryless nodes within the system of Kiick, since this is routine within NUMA systems.

Kiick fails to teach assigning the interrupts for the I/O devices to a first node having a cache, memory, and at least one processor and assigning the interrupt for the I/O device to the first node. Similarly, Kiick fails to teach repeating the assignment for the second and third nodes, respectively, if the first node to which the device is connected does not have a cache, memory, and at least one processor.

Examiner takes Official Notice that it would be obvious to assign the interrupt to a node which has a processor and memory, since these elements are required to service an interrupt (as evidenced by Agatsuma et al.; col. 4, line 64 - col. 5, line 9; col. 6, lines 15 - 22; Drottar et al.; col. 21, lines 15 - 22; col. 22, lines 44 - 58; col. 23, lines 22 – 30; and Fischer et al.; col. 9, lines 35 - 51). Examiner notes that all modern processor architectures contain a "memory" since they contain registers which are operated upon internally. Cache is old and notoriously well known in the art for providing improved performance in computing systems (as evidenced by Chi et al.; col. 2, lines 29 – 36; and Fischer et al.; col. 1, line 31 – col. 2, line 40).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to assign the interrupt for the I/O devices of Kiick to nodes which contain processors and memory, since both of these components are known in the prior art to be required in order to service an interrupt. It would have been obvious to use choose nodes which have cache, since it is well-established in this art that caching provides greater performance.

With regard to claim 8, Kiick teaches the system of claim 7, wherein the memory of each node that has memory is local to the node and remote to all other nodes (Figure 1, items 108A, 108B; paragraph 23 describes each domain having domain-specific memory (where a domain may be interpreted as a node, as described earlier in paragraph 23)), and the interrupt-assignor is to assign the interrupt for each I/O device to one of the plurality of nodes that has memory and at least one processor (where an interrupt-assignor may be interpreted as a dynamic interrupt distributor; Figure 2, item 210; paragraphs 25, 28; all nodes (items 102A, 102B are shown in Figure 1 to have memory and at least one processor).

With regard to claim 9, Kiick teaches the system of claim 8, wherein at least one of the I/O devices are performance critical, the interrupt-assignor further to assign the interrupt for each I/O device that is performance critical among the at least one processor of the node to which the interrupt has been assigned in a round-robin manner (Examiner interprets all of the I/O devices of the invention of Kiick to be performance critical, thereby necessitating the use of his invention to improve performance; Paragraphs 25, 26, 28).

With regard to claim 10, Kiick describes the system of claim 7, wherein, for each node that has processors, the interrupt-assignment software is further to dynamically modify assignments of the interrupts that are performance critical among the at least one processor of the node based on actual performance characteristics of the assignments. Examiner believes Applicant meant to refer to “the interrupt-assignor” instead of “interrupt-assignment software.” (Paragraphs 26, 28, 31).

With regard to claim 12, Kiick describes the system of claim 7, wherein the interrupt-assignor is further to dynamically modify assignments of the interrupts among the plurality of nodes based on actual performance characteristics of the assignments (Paragraphs 26, 28, 31).

Regarding claim 13, Kiick teaches wherein the interrupt-assignor (paragraph 34, where a interrupt-assignor may be interpreted as a dynamic interrupt distributor) is to give primary preference in assigning the interrupt for each I/O device to the node to which the I/O device is connected (paragraph 34, where a domain may be interpreted as a node) where the node to which the I/O device is connected has a cache (paragraph 10; Examiner interprets the processors within the processor complex (106A, 106B) as having on-chip cache), memory (Figure 1, items 108A, 108B) , and at least one processor (Figure 1, items 106A, 106B).

Regarding claim 14, Kiick teaches wherein each I/O device further has an interrupt service routine residing at one of the plurality of nodes, and the interrupt-assignor (paragraph 34, where a interrupt-assignor may be interpreted as a dynamic interrupt distributor) is to give secondary preference in assigning the interrupt for each I/O device to the node at which the interrupt service routine of the I/O device resides (paragraphs 28, 34; Examiner notes that paragraph 28 identifies re-assigning interrupts to be equivalent to re-assigning ISRs) where the node at which the interrupt service routine of the I/O device resides has a cache (paragraph 10; Examiner interprets the processors within the processor complex (106A, 106B) as having on-chip cache), memory (Figure 1, items 108A, 108B), and at least one processor ((Figure 1, items 106A, 106B).

With regard to claim 15, Kiick describes the system of claim 7, wherein the interrupt-assignor resides within one of the plurality of nodes (where an interrupt-assignor may be interpreted as a dynamic interrupt distributor; Figure 2, item 210; paragraph 28 describes a predetermined processor in a domain (node) dedicated to run the interrupt-assignor).

With regard to claim 16, Kiick teaches a computer-readable storage medium:

A computer readable medium;

Means in the medium for assigning interrupts for a plurality of input/output (I/O) devices (paragraph 28 describes a dynamic interrupt distributor embodied as a program module. Examiner identifies that a program module must be embodied on a computer readable medium in order to be useful, and therefore implicitly describes this limitation) among a plurality of nodes based on at least one factor selected from the set consisting of: the nodes to which the devices are connected, and the nodes at which interrupt service routines for the I/O devices reside (Paragraph 34 describes that interrupts should be assigned to the "closest" processors, and not across node boundaries. Examiner interprets this to mean the interrupts for the I/O devices should be assigned to nodes to which they are connected or to nodes where the ISRs for the said I/O devices reside.), where one or more of the nodes have processors and memory.

Kiick fails to teach where one or more of the nodes are memoryless.

Examiner takes Official Notice that nodes in a NUMA system may or may not contain memory. Rowlands et al. evidences this (col. 4, lines 1 – 2).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to provide memoryless nodes within the system of Kiick, since this is routine within NUMA systems.

Kiick fails to teach assigning the interrupts for the I/O devices to a first node having a cache, memory, and at least one processor and assigning the interrupt for the I/O device to the first node. Similarly, Kiick fails to teach repeating the assignment for the second and third nodes, respectively, if the first node to which the device is connected does not have a cache, memory, and at least one processor.

Examiner takes Official Notice that it would be obvious to assign the interrupt to a node which has a processor and memory, since these elements are required to service an interrupt (as evidenced by Agatsuma et al.; col. 4, line 64 - col. 5, line 9; col. 6, lines 15 - 22; Drottar et al.; col. 21, lines 15 - 22; col. 22, lines 44 - 58; col. 23, lines 22 – 30; and Fischer et al.; col. 9, lines 35 - 51). Examiner notes that all modern processor architectures contain a "memory" since they contain registers which are operated upon internally. Cache is old and notoriously well known in the art for providing improved performance in computing systems (as evidenced by Chi et al.; col. 2, lines 29 – 36; and Fischer et al.; col. 1, line 31 – col. 2, line 40).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to assign the interrupt for the I/O devices of Kiick to nodes which contain processors and memory, since both of these components are known in the prior art to be required in order to service an interrupt. It would have been obvious to use choose nodes which have cache, since it is well-established in this art that caching provides greater performance.

With regard to claim 17, Kiick teaches the medium of claim 16, wherein the means is for assigning the interrupts among the plurality of nodes further based on whether the nodes have processors and memories (Kiick describes assigning ISRs to processors which have associated memories; paragraph 14).

With regard to claim 18, Kiick describes the medium of claim 16, wherein the means, for each node having processors, is further for assigning the interrupts for the devices that are performance critical and that have been assigned to the node among the processors of the node in a round-robin manner (Examiner interprets all of the I/O devices of the invention of Kiick to be performance critical, thereby necessitating the use of his invention to improve performance; Paragraphs 25, 26, 28).

With regard to claim 19, Kiick describes the medium of claim 18, wherein the means, is further for dynamically modifying assignments of the interrupts among the nodes based on actual performance characteristics of the assignments, and, for each node having processors, for dynamically modifying assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node based on actual performance characteristics of the assignments (paragraphs 25, 28, 31).

With regard to claim 21, teaches describes a computer-readable storage medium comprising:

An interrupt-assignor (Figure 2, item 210; paragraph 28) to assign interrupts for a plurality of input/output (I/O) devices among a plurality of nodes based on at least one factor selected from the set consisting of:

The nodes to which the I/O devices are connected;

The nodes at which interrupt service routines for the I/O devices reside, where one or more of the nodes have processors and memory.

(Paragraph 34 describes that interrupts should be assigned to the "closest" processors, and not across node boundaries. Examiner interprets this to mean the interrupts for the I/O devices should be assigned to nodes to which they are connected or to nodes where the ISRs for the said I/O devices reside.).

Kiick fails to teach where one or more of the nodes are memoryless.

Examiner takes Official Notice that nodes in a NUMA system may or may not contain memory. Rowlands et al. evidences this (col. 4, lines 1 – 2).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to provide memoryless nodes within the system of Kiick, since this is routine within NUMA systems.

Kiick fails to teach assigning the interrupts for the I/O devices to a first node having a cache, memory, and at least one processor and assigning the interrupt for the I/O device to the first node. Similarly, Kiick fails to teach repeating the assignment for the second and third nodes, respectively, if the first node to which the device is connected does not have a cache, memory, and at least one processor.

Examiner takes Official Notice that it would be obvious to assign the interrupt to a node which has a processor and memory, since these elements are required to service an interrupt (as evidenced by Agatsuma et al.; col. 4, line 64 - col. 5, line 9; col. 6, lines 15 - 22; Drottar et al.; col. 21, lines 15 - 22; col. 22, lines 44 - 58; col. 23, lines 22 – 30; and Fischer et al.; col. 9, lines 35 - 51). Examiner notes that all modern processor architectures contain a "memory" since they contain registers which are operated upon internally. Cache is old and notoriously well known in the art for providing improved performance in computing systems (as evidenced by Chi et al.; col. 2, lines 29 – 36; and Fischer et al.; col. 1, line 31 – col. 2, line 40).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to assign the interrupt for the I/O devices of Kiick to nodes which contain processors and memory, since both of these components are known in the prior art to be required in order to service an interrupt. It would have been obvious to use choose nodes which have cache, since it is well-established in this art that caching provides greater performance.

With regard to claim 22, Kiick teaches the medium of claim 216, wherein the means is for assigning the interrupts among the plurality of nodes further based on whether the nodes have processors and memories (Kiick describes assigning ISRs to processors which have associated memories; paragraph 14).

With regard to claim 23, Kiick teaches the medium of claim 21, wherein the interrupt-assignor is to assign, for each node, the interrupts for the devices that are performance critical and that have been assigned to the node among the processors of the node in a round-robin manner (where the interrupt-assignor may be interpreted as a dynamic interrupt distributor; Examiner interprets all of the I/O devices of the invention of Kiick to be performance critical, thereby necessitating the use of his invention to improve performance; Paragraphs 25, 26, 28).

With regard to claim 24, Kiick teaches the medium of claim 23, wherein the interrupt-assignor is to dynamically modify assignments of the interrupts among the nodes based on actual performance characteristics of the assignments, and, for each node having processors, to dynamically modify assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node based on actual performance characteristics of the assignments (where an interrupt-assignor may be interpreted as a dynamic interrupt distributor; paragraphs 25, 28, 31).

With regard to claim 25, Kiick teaches a method comprising:

Assigning interrupts for a plurality of input/output (I/O) devices among a plurality of nodes based on at least one factor selected from the set consisting of: the nodes to which the I/O devices are connected; and the nodes at which interrupt service routines for the I/O devices reside, where one or more of the nodes have processors and memory (Paragraph 34 describes that interrupts should be assigned to the "closest" processors, and not across node boundaries. Examiner interprets this to mean the interrupts for the I/O devices should be assigned to nodes to which they are connected or to nodes where the ISRs for the said I/O devices reside.);

For each node of the system, assigning the interrupts for the devices that are performance critical and that have been assigned to the node among the processors of the node in a round-robin manner (Examiner interprets all of the I/O devices of the invention of Kiick to be performance critical, thereby necessitating the use of his invention to improve performance; Paragraphs 25, 26, 28);

Dynamically modifying assignments of the interrupts among the nodes of the system based on actual performance characteristics of the assignments (paragraphs 25, 28, 31);

For each node of the system, dynamically modifying assignments of the interrupts that are performance critical and that have been assigned to the node among the processors of the node based on actual performance characteristics of the assignments (paragraphs 25, 28, 31).

Kiick fails to teach where one or more of the nodes are memoryless.

Examiner takes Official Notice that nodes in a NUMA system may or may not contain memory. Rowlands et al. evidences this (col. 4, lines 1 – 2).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to provide memoryless nodes within the system of Kiick, since this is routine within NUMA systems.

Kiick fails to teach assigning the interrupts for the I/O devices to a first node having a cache, memory, and at least one processor and assigning the interrupt for the I/O device to the first node. Similarly, Kiick fails to teach repeating the assignment for the second and third nodes, respectively, if the first node to which the device is connected does not have a cache, memory, and at least one processor.

Examiner takes Official Notice that it would be obvious to assign the interrupt to a node which has a processor and memory, since these elements are required to service an interrupt (as evidenced by Agatsuma et al.; col. 4, line 64 - col. 5, line 9; col. 6, lines 15 - 22; Drottar et al.; col. 21, lines 15 - 22; col. 22, lines 44 - 58; col. 23, lines 22 – 30; and Fischer et al.; col. 9, lines 35 - 51). Examiner notes that all modern processor architectures contain a "memory" since they contain registers which are operated upon internally. Cache is old and notoriously well known in the art for providing improved performance in computing systems (as evidenced by Chi et al.; col. 2, lines 29 – 36; and Fischer et al.; col. 1, line 31 – col. 2, line 40).

Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to assign the interrupt for the I/O devices of Kiick to nodes which contain processors and memory, since both of these components are known in the prior art to be required in order to service an interrupt. It would have been obvious to use choose nodes which have cache, since it is well-established in this art that caching provides greater performance.

With regard to claim 26, Kiick teaches the method of claim 25, wherein assigning the interrupts for the plurality of I/O devices among the plurality of nodes of the system comprises, for each I/O device:

Where the node (Figure 1, items 102A, 102B) to which the I/O device (Figure 1, items 110A, 110B) is connected has a cache (Paragraph 10), memory (Figure 1, items 108A, 108B), and at least one processor (Figure 1, items 106A, 106B), assigning the interrupt for the I/O device to the node to which the I/O device is connected;

Otherwise, where the node at which the interrupt service routine for the I/O device resides has memory and at least one processor, assigning the interrupt for the I/O device to the node at which the interrupt service routine for the I/O device resides (Paragraph 34 describes that interrupts should be assigned to the "closest" processors, and not across node boundaries. Examiner interprets this to mean the interrupts for the I/O devices should be assigned to nodes to which they are connected or to nodes where the ISRs for the said I/O devices reside.).

With regard to claim 27, Kiick teaches the method of claim 25, wherein assigning the interrupts for the plurality of I/O devices among the plurality of nodes of the NUMA system further comprises, for each I/O device, otherwise, assigning the interrupt for the I/O device to one of the nodes having memory and at least one processor (Paragraph 23 describes each node containing memory (Figure 1, items 108A, 108B), and at least one processor (Figure 1, items 106A, 106B); paragraph 26).

With regard to claim 28, Kiick teaches the method of claim 25, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises:

Measuring responsiveness of the node in processing the interrupt (paragraphs 27 – 30);

Kiick fails to explicitly teach assigning the interrupt to the node at which the interrupt service routine for the I/O device resides; measuring responsiveness of the node at which the interrupt service routine for the I/O device resides in processing the interrupt; and where the responsiveness of the node to which the I/O device is connected is better than the responsiveness of the node at which the interrupt service routine for the I/O device resides, reassigning the interrupt to the node to which the I/O device is connected.

Kiick does, however, teach that the dynamic interrupt distributor should be aware of the system architecture, and re-assign interrupts to the "closest" processors (paragraph 34). Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to give preference in assigning interrupts to nodes at which the ISR for the device resides, or at which the I/O device itself resides, and then to re-assign interrupts accordingly to which configuration produced better performance as described in paragraphs 28 – 30).

With regard to claim 29, Kiick teaches the method of claim 25, wherein dynamically modifying the assignments of the interrupts among the nodes of the system comprises, for each assignment of an interrupt for an I/O device to a node, where the node is that at which the interrupt service routine for the I/O device resides:

Measuring responsiveness of the node in processing the interrupt (paragraphs 27 – 30);

Kiick fails to explicitly teach assigning the interrupt to the node to which the I/O device is connected; measuring responsiveness of the node to which the I/O device is connected in processing the interrupt; and where the responsiveness of the node at which the interrupt service routine for the I/O device is connected is better than the responsiveness of the node to which the I/O device is connected, reassigning the interrupt to the node at which the interrupt service routine for the I/O device resides.

Kiick does, however, teach that the dynamic interrupt distributor should be aware of the system architecture, and re-assign interrupts to the "closest" processors (paragraph 34). Therefore, it would have been obvious to one of ordinary skill in this art at the time of invention by Applicant to give preference in assigning interrupts to nodes at which the I/O device is connected, or at which the ISR for the /O device itself resides, and then to re-assign interrupts accordingly to which configuration produced better performance as described in paragraphs 28 – 30).

With regard to claim 30, Kiick teaches the method of claim 25, wherein for each node of the system having memory, dynamically modifying the assignments of the interrupts that are performance critical and that have been assigned to the node among the process ors of the node comprises:

Measuring the responsiveness of the processors of the node in processing the interrupts assigned thereto (paragraphs 27, 28, 35);

Where a differential between a best responsiveness and a worst responsiveness is greater than a threshold (paragraph 28; where a threshold may be interpreted as a "large enough difference");

Reassigning at least one of the interrupts assigned to the processor having the worst responsiveness to the processor having the best responsiveness (paragraphs 27 – 30, 35).

Response to Arguments

Applicant's arguments with respect to claim 1 – 30 have been considered but are moot in view of the new ground(s) of rejection in view of what is old and well known in the art as evidenced by Fischer et al. (U.S. 6,438,672), Drottar et al. (U.S. 6,170,025), Agatsuma et al. (U.S. 7,237,099), and Chi et al. (U.S. 6,209,086).

Examiner takes Official Notice that it would be obvious to assign the interrupt to a node which has a processor and memory, since these elements are required to service an interrupt (as evidenced by Agatsuma et al.; col. 4, line 64 - col. 5, line 9; col. 6, lines 15 - 22; Drottar et al.; col. 21, lines 15 - 22; col. 22, lines 44 - 58; col. 23, lines 22 – 30; and Fischer et al.; col. 9, lines 35 - 51). Examiner notes that all modern processor architectures contain a "memory" since they contain registers which are operated upon internally. Cache is old and notoriously well known in the art for providing improved performance in computing systems (as evidenced by Chi et al.; col. 2, lines 29 – 36; and Fischer et al.; col. 1, line 31 – col. 2, line 40).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Matthew D. Spittle whose telephone number is (571) 272-2467. The examiner can normally be reached on Monday - Friday, 8 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mark Rinehart can be reached on 571-272-3632. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Matthew Spittle

MDS

MARK H. RINEHART
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100